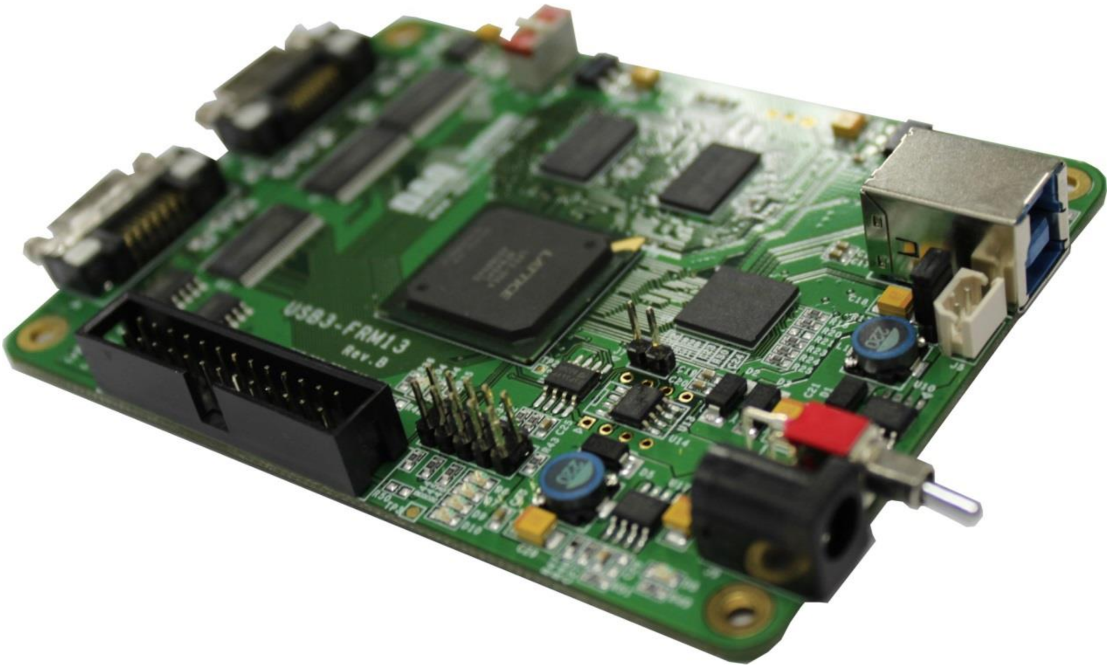
USB3-FRM13_K

API Manual

Version 1.0



Information furnished by DAQ SYSTEM is believed to be accurate and reliable, However, no responsibility is assumed by DAQ SYSTEM for its use, nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or copyrights of DAQ SYSTEM.

The information in this document is subject to change without notice and no part of this document may be copied or reproduced without the prior written consent.

Contents

Board Level API Functions

LVDS(Camera Link) API Functions

UART API Functions

DIO Input/Output API Functions

Multi-Board LVDS(Camera Link) API Functions

Multi-Board UART API Functions

Multi-Board DIO Input/Output API Functions

Board Level API Functions

Overview

int	**OpenDAQDevice (void)**
BOOL	**ResetBoard (int nBoard)**
BOOL	**CloseDAQDevice (void)**
int	**GetBoardNum (void)**
BOOL	**IsConnected(void)**

OpenDAQDevice

It opens a device. You may call this function at the very first time you run the program and some suspicious operation.

int **OpenDAQDevice (void)**

Parameters: None .

Return Value:

If the function succeeds, it returns the number of boards which were detected.

(Up to 4 for multi-board)

If the function fails, the return value is -1, it means there is no device in the system.

ResetBoard

It initializes a device at currently equipped system (PC).

BOOL **ResetBoard (int nBoard)**

Parameters:

nBoard : The Board number is set by dip switch.

Return Value:

It returns TRUE in case of the success of reset and initialization.

If you get FALSE you should not call any API functions with the board and call the **CloseDAQDevice()** instead.

CloseDAQDevice

The CloseDAQDevice function closes all opened devices (boards). If use of device is finished, it can certainly close a device for making it other programs so as usable.

BOOL CloseDAQDevice (void)

Parameters: None.

Return Value:

If the function fail to close, it returns “FALSE”.

If the function succeed to close, it returns “TRUE”.

GetBoardNum

This function returns currently detected board number in the system.

int GetBoardNum (void)

Parameters: None

Return Value:

The number of detected boards, The Board number is set by dip switch.

IsConnected

This function tells whether the USB is connected.

BOOL IsConnected (void)

Parameters: None.

Return Value:

If the function fail to close, it returns “FALSE”.

If the function succeed to close, it returns “TRUE”.

LVDS(Camera Link) API Functions

Overview

BOOL	**LVDS_Init (void)**
BOOL	**LVDS_Start (void)**
BOOL	**LVDS_GetFrame (DWORD* nCnt, unsigned char* buf)**
BOOL	**LVDS_Close (void)**
BOOL	**LVDS_GetResolution (DWORD *xRes, DWORD *yRes)**
BOOL	**LVDS_Stop (void)**
BOOL	**LVDS_SetDataMode (int nMode)**
BOOL	**LVDS_GetVersion (int *nFpgaVer, int *nFirmVer)**
DWORD	**LVDS_GetError (DWORD *dwStatuse)**
BOOL	**LVDS_BufferFlush (void)**
BOOL	**LVDS_SetDeUse (BOOL bUse)**
BOOL	**LVDS_SetHsPol (BOOL bPol)**
BOOL	**LVDS_CameraMode (int nMode)**
BOOL	**LVDS_SetReferrenceClock (int nClock)**
BOOL	**LVDS_ConfigureCc (DWORD dwCFG)**
BOOL	**LVDS_CcOutput (DWORD dwCC)**
BOOL	**LVDS_SetLineCount (DWORD dwCount)**
BOOL	**LVDS_SetPageStart (DWORD dwEvent)**
BOOL	**LVDS_SetPageDelay (DWORD dwCount)**
BOOL	**LVDS_ConfigureTrig1 (DWORD dwEvent, DWORD dwDelay, DWORD dwWidth, DWORD dwBlank)**
BOOL	**LVDS_ConfigureTrig2 (DWORD dwEvent, DWORD dwDelay, DWORD dwWidth, DWORD dwBlank)**
BOOL	**LVDS_SetRolling (BOOL bRolling)**
BOOL	**LVDS_SetPoCLDelay (int nMode)**
BOOL	**LVDS_ExtTrigEnable (BOOL bEn)**
BOOL	**LVDS_ExtTrigInv (BOOL bInv)**
BOOL	**LVDS_ExtTriConfigure (DWORD dwDelay, DWORD dwWidth, DWORD dwBlank)**
BOOL	**LVDS_EncAInv (BOOL bInv)**
BOOL	**LVDS_EncBInv (BOOL bInv)**
BOOL	**LVDS_EncZInv (BOOL bInv)**
BOOL	**LVDS_LineTrigInv (BOOL bInv)**
BOOL	**LVDS_PhaseTrigInv (BOOL bInv)**
BOOL	**LVDS_TrigOutInv (int nBoard, int nInv)**

LVDS_Init

This function initializes resources used for the LVDS sub-system, for example interrupt and LVDS control register.

BOOL LVDS_Init (int nBoard)

Parameters: None

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_Start

This function starts receiving frame data. After calling this function, you can check whether the data is complete by calling the LVDS_GetFrame function.

BOOL LVDS_Start (int nBoard)

Parameters: None

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_GetFrame

This function checks whether the frame data is complete, and if it is, retrieves the frame data. At this time, the size of the buffer to receive data must be informed.

BOOL LVDS_GetFrame (DWORD* nCnt, unsigned char* buf)

Parameters:

*nCnt : It is the address which contains the number of data to be received in byte size. Specifies the size buffer when the function is called, and read the values of the variables after a call to find out how many actually read. The data size is in bytes.

*buf : The buffer address.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, check the values of the size that you want to read nCnt.

(Note) If the frame data is not completed, FALSE is returned immediately and the return occurs with the nCnt value set to 0.

LVDS_Close

This function releases all resource were used for LVDS function.

BOOL LVDS_Close (void)

Parameters: None.

Return Value :

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_GetResolution

This function gets currently configured camera's frame resolution

BOOL LVDS_GetResolution (DWORD *xRes, DWORD *yRes)

Parameters:

*xRes : Address pointer to receive horizontal Camera resolution

*yRes : Address pointer to receive vertical Camera resolution

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_Stop

This function stops the frame data capture.

BOOL LVDS_Stop (void)

Parameters: None.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_SetDataMode

This function sets image pixel data mode.

BOOL LVDS_SetDataMode (int nMode)

Parameters:

nMode : If "0" is 8-bit mode,
"1" is 16-bit mode, "2" is 24-bit mode,
"3" is 32-bit mode, and "4" is 16-bit YUV (convert mode).

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_GetVersion

This function gets FPGA version.

BOOL LVDS_GetVersion (int *nVersion)

Parameters:

*nVersion : The pointer of the FPGA version.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_GetError

This function gets the error state.

DWORD LVDS_GetError (DWORD *dwStatus)

Parameters:

*dwStatus : "1" : Overflow error
"2" : Unread Error
"4" : Size Error

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_BufferFlush

This function initializes the buffer.

BOOL LVDS_BufferFlush (void)

Parameters: None.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_SetDeUse

This function sets a Data Valid or Horizontal Synchronization signal.

BOOL LVDS_SetDUse (BOOL bUse)

Parameters:

bUse : If "True", DVAL (Data Validation) is used.
If "False", HSYNC (Horizontal Synchronization) is used.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_SetHsPol

This function selects the horizontal signal (HSYNC: Horizontal Synchronization) signal line.

BOOL LVDS_SetHsPol (BOOL bPol)

Parameters:

bPol : If "True", Normal HSYNC is used,
If "False", Inverse HSYNC is used.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_CameraMode

This function selects whether the camera mode is Area Line Scan Camera or Line Scan Camera.

BOOL LVDS_CameraMode (int nMode)

Parameters:

nMode : If "0", Area Scan Camera is used (Default),
If "Others", Line Scan Camera is used.

Return Value:

If the function call fails, it returns "FALSE".
If the function call succeeds, it returns "TRUE".

LVDS_SetReferenceClock

This function selects which reference clock to use.

BOOL LVDS_SetReferenceClock (int nClcok)

Parameters:

nClock : "If it is 0", the internal clock in the frame grabber is used (Default)
If it is "Others", the external clock provided by the encoder or other board is used.

Return Value:

If the function call fails, it returns "FALSE".
If the function call succeeds, it returns "TRUE".

LVDS_ConfigureCc

This function selects the configuration value of the corresponding bit.

BOOL LVDS_ConfigureCc (DWORD dwCFG)

Parameters:

dwCFG : bit0(CC1 configure) = "0" : digital out1 / "1": alternate (Trigger1 output)
bit1(CC2 configure) = "0" : digital out2 / "1": alternate (Trigger2 output)
bit2(CC3 configure) = "0" : digital out3 / "1": (Digital output)
bit3(CC4 configure) = "0" : digital out4 / "1": alternate (Reference clock output)
others : Reserved

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_CcOutput

This function selects the CC (Camera Control) to use.

BOOL LVDS_CcOutput (DWORD dwCC)

Parameters:

dwCC :	"bit0" :	CC1 out
	"bit1" :	CC2 out
	"bit2" :	CC3 out
	"bit3" :	CC4 out
	"others" :	Reserved

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_SetLineCount

This function selects the number of lines needed to get one image in one page.

BOOL LVDS_SetLineCount (DWORD dwCount)

Parameters:

dwCount : 1 ~65535

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_SetPageStart

This function selects the page initial event.

BOOL LVDS_SetPageStart (DWORD dwEvent)

Parameters:

dwEvent : "0" : Continuous (Free Running)
"1" : Rising edge in Page Trigger input
"2" : Rising edge on encoder z phase

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_SetPageDelay

This function selects the number of clocks needed before getting one image from one page.

BOOL LVDS_SetPageDelay (DWORD dwCount)

Parameters:

dwCount : 1 ~ 15

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_ConfigureTrig1

This function selects trigger 1's input mode selection, output delay, output width, and output blank.

BOOL LVDS_ConfigureTrig1 (DWORD dwEvent, DWORD dwDelay,
DWORD dwWidth, DWORD dwBlank)

Parameters:

dwEvent : "0" : Continuous (Free running without any condition)
"1" : Rising edge on Line Trigger input
"2" : Rising edge on encoder A phase
"3" : Rising edge on encoder B phase
"4" : Rising edge encoder up clock using A and B phase
"5" : Rising edge encoder down clock A and B phase

dwDelay : number of reference clocks required before output Trig1 goes high

dwWidth : number of reference clocks required for output Trig1 to go high

dwBlank : number of reference clocks required before output Trig1 returns from delay

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_ConfigureTrig2

This function selects trigger 2's input mode selection, output delay, output width, and output blank.

BOOL LVDS_ConfigureTrig2 (DWORD dwEvent, DWORD dwDelay,
DWORD dwWidth, DWORD dwBlank)

Parameters:

dwEvent : "0" : Continuous (Free running without any condition)
"1" : Rising edge on Line Trigger input
"2" : Rising edge on encoder A phase
"3" : Rising edge on encoder B phase
"4" : Rising edge encoder up clock using A and B phase
"5" : Rising edge encoder down clock A and B phase
dwDelay : number of reference clocks required before output Trig1 goes high
dwWidth : number of reference clocks required for output Trig1 to go high
dwBlank : number of reference clocks required before output Trig1 returns from delay

Return Value:

If the function call fails, it returns "FALSE".
If the function call succeeds, it returns "TRUE".

LVDS_SetRolling

This function updates LVDS frame data and image data without using the GetFrame function.

BOOL LVDS_SetRolling (BOOL bRolling)

Parameters:

bRolling : If "1", the frame data is updated regardless of the GetFrame function.

Return Value:

If the function call fails, it returns "FALSE".
If the function call succeeds, it returns "TRUE".

LVDS_SetPoCLDelay

This function can select the delay time when using Power over Camera Link (PoCL) power. (DEFAULT is 0.5sec.)

BOOL LVDS_SetPoCLDelay (int nMode)

Parameters:

nMode : "0" : 0.5sec, "1" : 1sec, "2" : 1.5sec, "3" : 2sec
"4" : 2.5sec, "5" : 3sec, "6" : 3.5sec, "7" : 4sec

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_ExtTrigEnable

This function selects an external trigger signal.

BOOL LVDS_ExtTrigEnable (BOOL bEn)

Parameters:

bEn : If "True", an external trigger signal is used and
If "False", internal trigger signal is used.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_ExtTrigInv

This function inverts the external trigger signal.

BOOL LVDS_ExtTrigInv (BOOL bInv)

Parameters:

bInv : If it is "True", the external trigger signal is inversely converted and used.
If it is "False", the external trigger signal is used as it is.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_ExtTrigConfigure

This function configures the external trigger signal.

BOOL LVDS_ExtTrigConfigure (DWORD dwDelay, DWORD dwWidth, DWORD dwBlank)

Parameters:

dwDelay : the number of reference clocks required before the external trigger signal goes high.

dwWidth : the number of reference clocks required for the external trigger signal to be high.

dwBlank : The number of reference clocks required before the external trigger signal returns from the delay.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_EncAInv

This function inverts the encoder A signal.

BOOL LVDS_EncAInv (BOOL bInv)

Parameters:

bInv : If it is "True", the Encoder A signal is inversely transformed and used.
If it is "False", Encoder A signal is used as it is.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_EncBInv

This function inverts the encoder B signal.

BOOL LVDS_EncBInv (BOOL bInv)

Parameters:

bInv : If it is "True", the Encoder B signal is inversely transformed and used.
If it is "False", Encoder B signal is used as it is.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_EncZInv

This function inverts the encoder Z signal..

BOOL LVDS_EncZInv (BOOL bInv)

Parameters:

bInv : If it is "True", the Encoder Z signal is inversely transformed and used.
If it is "False", Encoder Z signal is used as it is.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_LineTrigInv

This function inverts the Line Trigger signal.

BOOL LVDS_LineTrigInv (BOOL bInv)

Parameters:

bInv : If "True", the Line Trigger signal is inversely converted and used.
If it is "False", the Line Trigger signal is used as it is.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_PhaseTrigInv

This function inverts the Phase Trigger signal.

BOOL LVDS_PhaseTrigInv (BOOL bInv)

Parameters:

bInv : If "True", the Phase Trigger signal is inversely converted and used.
If it is "False", the Phase Trigger signal is used as it is.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_TrigOutInv

This function inverts the CC output waveform.

BOOL LVDS_TrigOutInv (int nInv)

Parameters:

nInv : "0" : CC1, CC2 Normal Signal
"1" : CC1 Inverse Signal
"2" : CC2 Inverse Signal
"Others" : CC1, CC2 Inverse Signal

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

UART API Functions

Overview

BOOL	**UART_Init (void)**
BOOL	**UART_GetData (DWORD* nCnt, unsigned char* buf)**
BOOL	**UART_SendData (DWORD* nCnt, unsigned char* buf)**
BOOL	**UART_Close (void)**
BOOL	**UART_SetBaud (DWORD nBaud)**
BOOL	**UART_BufferFlush (void)**

UART_Init

This function initializes resources used for the UART sub-system, for example interrupt and UART control register.

BOOL UART_Init (void)

Parameters: None.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

UART_GetData

This function receives characters through the differential UART.

BOOL UART_GetData (DWORD* nCnt, unsigned char* buf)

Parameters:

*nCnt : The address which contains the number of characters to be received.

The maximum number of characters to be received is limited to 4Kbyte(4096).

*buf : The buffer address.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

UART_SendData

This function sends characters through the differential UART.

BOOL UART_SendData (DWORD* nCnt, unsigned char* buf)

Parameters:

*nCnt : The address which contains the number of characters to be sent.
The maximum number of characters to be sent is limited to 4K byte(4096).

*buf : The buffer address.

Return Value:

If the function call fails, it returns “FALSE”.

If the function call succeeds, it returns “TRUE”.

UART_Close

This function releases all resource were used for UART function.

BOOL UART_Close (void)

Parameters: None.

Return Value:

If the function call fails, it returns “FALSE”.

If the function call succeeds, it returns “TRUE”.

UART_SetBaud

This function sets the UART Baud.

BOOL UART_SetBaud (DWORD nBaud)

Parameters:

nBaud : 0: 9600, 1: 19200, 2: 38400, 3:57600, 4:115200

Return Value:

If the function call fails, it returns “FALSE”.

If the function call succeeds, it returns “TRUE”.

UART_BufferFlush

This function clears the UART Rx buffer.

BOOL UART_BufferFlush (void)

Parameters: None.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

Digital Input/Output API Functions

Overview

DWORD	**DIO_Read (void)**
BOOL	**DIO_Write (DWORD dwVal)**

DIO_Read

This function reads the value of the input port.

DWORD DIO_Read (void)

Parameters: None.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

DIO_Write

This function outputs the desired camera control value to the output port. (refer to camera specifications)

BOOL DIO_Write (DWORD dwVal)

Parameters:

dwVal : The value to be written to the port.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

Multi Board support APIs

In case of single board API, only one board is used in the installed system. However, in a system with two or more boards installed (up to 4 supported), multiple APIs must be used. Multi board API is only available for FPGA version #2 or higher boards.

Multi Board LVDS(Camera Link) APIs

Overview

BOOL	**IsConnected_Mul (int nBoard)**
BOOL	**LVDS_Init_Mul (int nBoard)**
BOOL	**LVDS_Check_Mul (int nBoard)**
BOOL	**LVDS_Start_Mul (int nBoard)**
BOOL	**LVDS_GetFrame_Mul (int nBoard, DWORD* nCnt, unsigned char* buf)**
BOOL	**LVDS_GetFrameCount_Mul (int nBoard, *nCount)**
BOOL	**LVDS_Close_Mul (int nBoard)**
BOOL	**LVDS_GetResolution_Mul (int nBoard, DWORD *xRes, DWORD *yRes)**
BOOL	**LVDS_Stop_Mul (int nBoard)**
BOOL	**LVDS_SetDataMode_Mul (int nBoard, int nMode)**
BOOL	**LVDS_GetVersion_Mul (int nBoard, int *nVersion)**
BOOL	**LVDS_GetError_Mul (int nBoard, DWORD *dwStatus)**
BOOL	**LVDS_BufferFlush_Mul (int nBoard)**
BOOL	**LVDS_SetDeUse_Mul (int nBoard, BOOL bUse)**
BOOL	**LVDS_SetHsPol_Mul (int nBoard, BOOL bPol)**
BOOL	**LVDS_CameraMode_Mul (int nBoard, int nMode)**
BOOL	**LVDS_SetReferrenceClock_Mul (int nBoard, int nClock)**
BOOL	**LVDS_ConfigureCc_Mul (int nBoard, DWORD dwCFG)**
BOOL	**LVDS_CcOutput_Mul (int nBoard, DWORD dwCC)**
BOOL	**LVDS_SetLineCount_Mul (int nBoard, DWORD dwCount)**
BOOL	**LVDS_SetPageStart_Mul (int nBoard, DWORD dwEvent)**
BOOL	**LVDS_SetPageDelay_Mul (int nBoard, DWORD dwCount)**
BOOL	**LVDS_ConfigureTrig1_Mul (int nBoard, DWORD dwEvent, DWORD dwDelay, DWORD dwWidth, DWORD dwBlank)**
BOOL	**LVDS_ConfigureTrig2_Mul (int nBoard, DWORD dwEvent, DWORD dwDelay, DWORD dwWidth, DWORD dwBlank)**
BOOL	**LVDS_SetRolling_Mul (int nBoard, BOOL bRolling)**
BOOL	**LVDS_SetPoCLDelay_Mul (int nBoard, int nMode)**
BOOL	**LVDS_ExtTrigEnable_Mul (int nBoard, BOOL bEn)**

BOOL LVDS_ExtTrigInv_Mul (int nBoard, BOOL bInv)

BOOL LVDS_ExtTriConfigure_Mul (int nBoard, DWORD dwDelay,
DWORD dwWidth, DWORD dwBlank)

BOOL LVDS_EncAInv_Mul (int nBoard, BOOL bInv)

BOOL LVDS_EncBInv_Mul (int nBoard, BOOL bInv)

BOOL LVDS_EncZInv_Mul (int nBoard, BOOL bInv)

BOOL LVDS_LineTrigInv_Mul (int nBoard, BOOL bInv)

BOOL LVDS_PhaseTrigInv_Mul (int nBoard, BOOL bInv)

BOOL LVDS_TrigOutInv_Mul (int nBoard, int nInv)

IsConnected_Mul

This function tells whether the USB is connected.

BOOL IsConnected_Mul (int nBoard)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_Init_Mul

This function initializes resources used for the LVDS sub-system, for example interrupt and LVDS control register.

BOOL LVDS_Init_Mul (int nBoard)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_Start_Mul

This function starts receiving frame data. After calling this function, you can check whether the data is complete by calling the LVDS_GetFrame function.

BOOL LVDS_Start_Mul (int nBoard)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_GetFrame_Mul

This function checks whether the frame data is complete, and if it is, retrieves the frame data. At this time, the size of the buffer to receive data must be informed.

BOOL LVDS_GetFrame_Mul (int nBoard, DWORD* nCnt, unsigned char* buf)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

*nCnt : It is the address which contains the number of data to be received in byte size. Specifies the size buffer when the function is called, and read the values of the variables after a call to find out how many actually read. The data size is in bytes.

*buf : The buffer address.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, check the values of the size that you want to read nCnt.

(Note) If the frame data is not completed, FALSE is returned immediately and the return occurs with the nCnt value set to 0.

LVDS_Close_Mul

This function releases all resource were used for LVDS function.

BOOL LVDS_Close (int nBoard)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

Return Value:

If the function call fails, it returns “FALSE”.

If the function call succeeds, it returns “TRUE”.

LVDS_GetResolution_Mul

This function gets currently configured camera’s frame resolution

BOOL LVDS_GetResolutuion_Mul (int nBoard, DWORD *xRes, DWORD *yRes)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

*xRes : Address pointer to receive horizontal Camera resolution

*yRes : Address pointer to receive vertical Camera resolution

Return Value:

If the function call fails, it returns “FALSE”.

If the function call succeeds, it returns “TRUE”.

LVDS_Stop_Mul

This function stops the frame data capture.

BOOL LVDS_Stop_Mul (int nBoard)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

Return Value:

If the function call fails, it returns “FALSE”.

If the function call succeeds, it returns “TRUE”.

LVDS_SetDataMode_Mul

This function sets image pixel data mode.

BOOL LVDS_SetDataMode_Mul (int nBoard, int nMode)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

nMode : If “0” is 8-bit mode, “1” is 16-bit mode, “2” is 24-bit mode,
“3” is 32-bit mode, and “4” is 16-bit YUV (convert mode).

Return Value:

If the function call fails, it returns “FALSE”.

If the function call succeeds, it returns “TRUE”.

LVDS_GetVersion_Mul

This function gets FPGA version.

BOOL LVDS_GetVersion_Mul (int nBoard, int *nVersion)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

*nVersion : The pointer of the FPGA version.

Return Value:

If the function call fails, it returns “FALSE”.

If the function call succeeds, it returns “TRUE”.

LVDS_GetError_Mul

This function gets the error state.

DWORD LVDS_GetError_Mul (int nBoard, int nHz, int *nResX, int *nResY)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

*dwStatus : "1" : Overflow error
"2" : Unread Error
"4" : Size Error

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_BufferFlush_Mul

This function initializes the buffer.

BOOL LVDS_BufferFlush_Mul (int nBoard)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_SetDeUse_Mul

This function sets a Data Valid or Horizontal Synchronization signal.

BOOL LVDS_SetDUse_Mul (int nBoard, BOOL bUse)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

bUse : If "True", DVAL (Data Validation) is used.
If "False", HSYNC (Horizontal Synchronization) is used.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_SetHsPol_Mul

This function selects the horizontal signal (HSYNC: Horizontal Synchronization) signal line.

BOOL LVDS_SetHsPol_Mul (int nBoard, BOOL bPol)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

bPol : If "True", Normal HSYNC is used,
If "False", Inverse HSYNC is used.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_CameraMode_Mul

This function selects whether the camera mode is Area Line Scan Camera or Line Scan Camera.

BOOL LVDS_CameraMode_Mul (int nBoard, int nMode)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

nMode : If "0", Area Scan Camera is used (Default),
If "Others", Line Scan Camera is used.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_SetReferenceClock_Mul

This function selects which reference clock to use.

BOOL LVDS_SetReferenceClock_Mul (int nBoard, int nClcok)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

nClock : "If it is 0", the internal clock in the frame grabber is used (Default)
If it is "Others", the external clock provided by the encoder or other board is used.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_ConfigureCc_Mul

This function selects the configuration value of the corresponding bit.

BOOL LVDS_ConfigureCc_Mul (int nBoard, DWORD dwCFG)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

dwCFG : bit0(CC1 configure) = "0" : digital out1 / "1": alternate (Trigger1 output)
bit1(CC2 configure) = "0" : digital out2 / "1": alternate (Trigger2 output)
bit2(CC3 configure) = "0" : digital out3 / "1": (Digital output)
bit3(CC4 configure)= "0" : digital out4 / "1": alternate (Reference clock output)
others : Reserved

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_CcOutput_Mul

This function selects the CC (Camera Control) to use.

To use CC1, write 1, CC2 uses 2, CC3 uses 4, and CC4 uses 8.

BOOL LVDS_CcOutput_Mul (int nBoard, DWORD dwCC)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

dwCC : "bit0" : CC1 out
"bit1" : CC2 out
"bit2" : CC3 out
"bit3" : CC4 out
"others" : Reserved

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_SetLineCount_Mul

This function selects the number of lines needed to get one image in one page.

BOOL LVDS_SetLineCount_Mul (int nBoard, DWORD dwCount)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

dwCount : 1 ~65535

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_SetPageStart_Mul

This function selects the page initial event.

BOOL LVDS_SetPageStart_Mul (int nBoard, DWORD dwEvent)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

dwEvent : "0" : Continuous (Free Running)
"1" : Rising edge in Page Trigger input
"2" : Rising edge on encoder z phase

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_SetPageDelay_Mul

This function selects the number of clocks needed before getting one image from one page.

BOOL LVDS_SetPageDelay_Mul (int nBoard, DWORD dwCount)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

dwCount : 1 ~ 15

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_ConfigureTrig1_Mul

This function selects trigger 1's input mode selection, output delay, output width, and output blank.

BOOL LVDS_ConfigureTrig1_Mul (int nBoard, DWORD dwEvent,
DWORD dwDelay, DWORD dwWidth, DWORD dwBlank)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

dwEvent : "0" : Continuous (Free running without any condition)
"1" : Rising edge on Line Trigger input
"2" : Rising edge on encoder A phase
"3" : Rising edge on encoder B phase
"4" : Rising edge encoder up clock using A and B phase
"5" : Rising edge encoder down clock A and B phase

dwDelay : number of reference clocks required before output Trig1 goes high

dwWidth : number of reference clocks required for output Trig1 to go high

dwBlank : number of reference clocks required before output Trig1 returns from delay

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_ConfigureTrig2_Mul

This function selects trigger 2's input mode selection, output delay, output width, and output blank.

BOOL LVDS_ConfigureTrig2_Mul (int nBoard, DWORD dwEvent,
DWORD dwDelay, DWORD dwWidth, DWORD dwBlank)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

dwEvent : “0” : Continuous (Free running without any condition)
“1” : Rising edge on Line Trigger input
“2” : Rising edge on encoder A phase
“3” : Rising edge on encoder B phase
“4” : Rising edge encoder up clock using A and B phase
“5” : Rising edge encoder down clock A and B phase

dwDelay : number of reference clocks required before output Trig1 goes high

dwWidth : number of reference clocks required for output Trig1 to go high

dwBlank : number of reference clocks required before output Trig1 returns from delay

Return Value:

If the function call fails, it returns “FALSE”.
If the function call succeeds, it returns “TRUE”.

LVDS_SetTrigger_Mul

This function starts a Trigger.

BOOL LVDS_SetTrigger_Mul (int nBoard, BOOL bUse)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

bUse : If “True”, trigger is used.
If it is “False”, it maintains the Normal state.

Return Value:

If the function call fails, it returns “FALSE”.
If the function call succeeds, it returns “TRUE”.

LVDS_SetPoCLDelay_Mul

This function can select the delay time when using Power over Camera Link (PoCL) power. (DEFAULT is 0.5sec.)

BOOL LVDS_SetPoCLDelay_Mul (int nBoard, int nMode)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

nMode : "0" : 0.5sec, "1" : 1sec, "2" : 1.5sec, "3" : 2sec
"4" : 2.5sec, "5" : 3sec, "6" : 3.5sec, "7" : 4sec

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_ExtTrigEnable_Mul

This function selects an external trigger signal.

BOOL LVDS_ExtTrigEnable_Mul (int nBoard, BOOL bEn)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

bEn: If "True", an external trigger signal is used and
If "False", internal trigger signal is used.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_ExtTrigInv_Mul

This function inverts the external trigger signal.

BOOL LVDS_ExtTrigInv_Mul (int nBoard, BOOL bInv)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

bInv : If it is "True", the external trigger signal is inversely converted and used.
If it is "False", the external trigger signal is used as it is.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_ExtTrigConfigure_Mul

This function configures the external trigger signal.

BOOL LVDS_ExtTrigConfigure_Mul (int nBoard, DWORD dwDelay, DWORD dwWidth, DWORD dwBlank)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

dwDelay : the number of reference clocks required before the external trigger signal goes high.

dwWidth : the number of reference clocks required for the external trigger signal to be high.

dwBlank : The number of reference clocks required before the external trigger signal returns from the delay.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_EncAInv_Mul

This function inverts the encoder A signal.

BOOL LVDS_EncAInv_Mul (int nBoard, BOOL bInv)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

bInv : If it is "True", the Encoder A signal is inversely transformed and used.
If it is "False", Encoder A signal is used as it is.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_EncBInv_Mul

This function inverts the encoder B signal.

BOOL LVDS_EncBInv_Mul (int nBoard, BOOL bInv)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

bInv : If it is "True", the Encoder B signal is inversely transformed and used.
If it is "False", Encoder B signal is used as it is.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_EncZInv_Mul

This function inverts the encoder Z signal..

BOOL LVDS_EncZInv_Mul (int nBoard, BOOL bInv)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

bInv : If it is "True", the Encoder Z signal is inversely transformed and used.
If it is "False", Encoder Z signal is used as it is.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_LineTrigInv_Mul

This function inverts the Line Trigger signal.

BOOL LVDS_LineTrigInv_Mul (int nBoard, BOOL bInv)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

bInv : If "True", the Line Trigger signal is inversely converted and used.
If it is "False", the Line Trigger signal is used as it is.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_PhaseTrigInv_Mul

This function inverts the Phase Trigger signal.

BOOL LVDS_PhaseTrigInv_Mul (int nBoard, BOOL bInv)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

bInv : If "True", the Phase Trigger signal is inversely converted and used.
If it is "False", the Phase Trigger signal is used as it is.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

LVDS_TrigOutInv_Mul

This function inverts the CC output waveform.

BOOL LVDS_TrigOutInv_Mul (int nBoard, int nInv)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

nInv : "0" : CC1, CC2 Normal Signal
"1" : CC1 Inverse Signal
"2" : CC2 Inverse Signal
"Others" : CC1, CC2 Inverse Signal

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

Multi-Board UARTAPI Functions

Overview

BOOL	**UART_Init_Mul (int nBoard)**
BOOL	**UART_GetData_Mul (int nBoard, DWORD* nCnt, unsigned char* buf)**
BOOL	**UART_SendData_Mul (int nBoard, DWORD* nCnt, unsigned char* buf)**
BOOL	**UART_Close_Mul (int nBoard)**
BOOL	**UART_SetBaud_Mul (int nBoard, DWORD nBaud)**
BOOL	**UART_BufferFlush_Mul (int nBoard)**

UART_Init_Mul

This function initialize resources used for the UART sub-system, for example interrupt and UART control register.

BOOL UART_Init_Mul (int nBoard)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

Return Value:

If the function call fails, it returns "FALSE".
If the function call succeeds, it returns "TRUE".

UART_GetData_Mul

This function receives characters through the differential UART.

BOOL UART_GetData_Mul (int nBoard, DWORD* nCnt, unsigned char* buf)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.
*nCnt : The address which contains the number of characters to be received.
The maximum number of characters to be received is limited to 4Kbyte(4096).
*buf : The buffer address.

Return Value:

If the function call fails, it returns "FALSE".
If the function call succeeds, it returns "TRUE".

UART_SendData_Mul

This function sends characters through the differential UART.

BOOL UART_SendData_Mul (int nBoard, DWORD* nCnt, unsigned char* buf)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

*nCnt : The address which contains the number of characters to be sent.
The maximum number of characters to be sent is limited to 4K byte(4096).

*buf : The buffer address.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

UART_Close_Mul

This function releases all resource were used for UART function.

BOOL UART_Close_Mul (int nBoard)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

UART_SetBaud_Mul

This function sets UART Baud rates.

BOOL UART_SetBaud_Mul (int nBoard, DWORD nBaud)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

nBaud : 0: 9600, 1: 19200, 2: 38400, 3:57600, 4: 115200

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

UART_BufferFlush_Mul

This function flushes UART RX Buffer

BOOL UART_BufferFlush_Mul (int nBoard)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

Multi-board CC(Camera Control) API Functions

Overview

DWORD DIO_Read (int nBoard)

BOOL DIO_Write (int nBoard, DWORD dwVal)

DIO_Read

This function reads the value of the input port.

DWORD DIO_Read (int nBoard)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

DIO_Write

This function outputs the desired camera control value to the output port.
(Refer to camera specifications)

BOOL DIO_Write (int nBoard, DWORD dwVal)

Parameters:

nBoard : It informs a board number at currently equipped system.
The board number set up by DIP switch.

dwVal : The value to be written to the port.

Return Value:

If the function call fails, it returns "FALSE".

If the function call succeeds, it returns "TRUE".

Memo

Contact Point

Web sit : https://www.daqsystem.com

Email : postmaster@daqsystem.com

DAQ SYSTEM
Measurement & Automation